(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 930 564 A1

(12) EUROPEAN PATENT APPLICATION

published in accordance with Art. 158(3) EPC

(43) Date of publication:
21.07.1999 Bulletin 1999/29

(21) Application number: **98912729.5**

(22) Date of filing: **08.04.1998**

(51) Int. Cl.[6]: **G06F 9/30**, G06F 9/38, G06F 7/00

(86) International application number:
PCT/JP98/01626

(87) International publication number:
WO 98/45774 (15.10.1998 Gazette 1998/41)

(84) Designated Contracting States:
DE FR GB

(30) Priority: **08.04.1997 JP 8975197**

(71) Applicant:
Sony Computer Entertainment Inc.
Tokyo 107-0052 (JP)

(72) Inventor:
OKA, Masaaki
Sony Computer Entertainment Inc.
Minato-ku Tokyo 107-0052 (JP)

(74) Representative:
Pilch, Adam John Michael et al
D. YOUNG & CO.,
21 New Fetter Lane
London EC4A 1DA (GB)

(54) ARITHMETIC UNIT AND ARITHMETIC METHOD

(57) An arithmetic and logic unit (ALU) 330, a shift processing unit (SHT) 340 and a register unit (REG) 350 each caused to be of structure divided into, e.g., four sections can mutually transfer data through buses (BUS) 360, 370, 380 of 64 bits. Data of plural fields within word subject to operation inputted to the ALU 330 are exchanged as occasion demands by data exchange units EXCs 310, 320 provided between the buses 360, 370 and the ALU 330. Thus, operation function betweenn plural fields within the same word subject to operation can be realized by the number of steps lesser than that of the prior art.

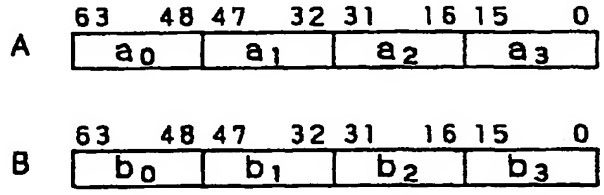

FIG.10

EP 0 930 564 A1

BNSDOCID: <EP___0930564A1_I_>

Description

Technical Field

[0001] This invention relates to an arithmetic apparatus and an arithmetic method for carrying out arithmetic and logical operation using CPU.

Background Art

[0002] Among CPUs (Central Processing Units) which are arithmetic units (arithmetic and logic units) used for computer, etc., there are some arithmetic units having a group of instructions called multimedia instruction (hereinafter referred to as MM instruction or simply referred to as instruction). This MM instruction serves to divide area of arithmetic (computing) element that CPU has to execute plural operations at the same time.

[0003] An example of the configuration of a conventional CPU is shown in FIG. 1. This conventional CPU comprises an arithmetic and logic unit (ALU) 130 serving as arithmetic and logic means for executing data processing, a shift processing unit (SHT) 140 serving as shift processing means for shifting data in left and right directions, and a register unit (REG) 150 such as accumulator, etc., wherein those units are connected to, e.g., buses 160, 170, 180 of 64 bits to mutually transfer data.

[0004] FIG. 2 shows multiplication by multiplier of 64 bits x 64 bits in the above-described conventional CPU. Namely, word s*t of 128 bits, which is product of word s of 64 bits of register A and word t of 64 bits of register B, is generated and is stored into register C.

[0005] FIG. 3 shows the state where the above-mentioned 64 bit words s and t are respectively divided into four fields to form respective four bit fields to carry out multiplication of bits of ack (acknowledge) fields, i.e., 16 bits x 16 bits. Namely, s0*t0, s1*t1, s2*t2 and s3*t3 respectively consisting of 32 bits which are products of respective 16 bits s0, s1, s2, s3 of the register A and respective 16 bits to, t1, t2, t3 of the register B are generated and are stored into the register C.

[0006] Such four parallel multiplication can be realized by quartering the multiplier that CPU has to constitute multipliers of four parallel. In addition, similarly to the above, adder that CPU has may be also quartered to constitute four parallel adders.

[0007] FIG. 4 shows addition by adder of 128 bits + 128 bits in the above-described conventional CPU. Namely, 128 bits s+t which are sum of respective 32 bits s of register A and respective 32 bits t of register B are generated and are stored into register C.

[0008] FIG. 5 shows the state where the above-mentioned respective words are quartered to carry out additions of respective 32 bits + respective 32 bits. Namely, s0+t0, s1+t1, s2+t2 and s3+t3 respectively consisting of 32 bits which are sums of respective 32 bits s0, s1, s2, s3 of the register A and respective 32 bits t0, t1, t2, t3 of the register B are generated and are stored into the register C.

[0009] When data width subject to operation is about 16 bits or 32 bits as stated above, if parallel arithmetic (computing) elements constituted by dividing single arithmetic (computing) element are used, it is possible to carry out arithmetic processing at a high speed. Instructions for carrying out parallel operation shown in FIGS. 3 and 5 are a portion of multi-media (MM) instructions used therefor.

[0010] A more practical example of conventional parallel operation using MM instruction is indicated below.

[0011] Initially, explanation will be given in connection with the case where n simultaneous linear equations as indicated by the following equation (1) are solved by using the Cramer's formula.

$$a_{00}X_0 + a_{01}X_1 + \cdots + a_{0n}X_n = b_0 \quad (1)$$

$$a_{10}X_0 + a_{11}X_1 + \cdots + a_{1n}X_n = b_1 \cdots$$

$$a_{n0}X_0 + a_{n1}X_1 + \cdots + a_{nn}X_n = b_n$$

BNSDOCID: <EP___0930564A1_I_>

$$Xj = \begin{vmatrix} a_{00} \cdots & b_0 \cdots & a_{0n} \\ \vdots & \vdots & \vdots \\ a_{n0} \cdots & b_n \cdots & a_{nn} \end{vmatrix} \Big/ \begin{vmatrix} a_{00} & \cdots\cdots & a_{0n} \\ \vdots & & \vdots \\ a_{n0} & \cdots\cdots & a_{nn} \end{vmatrix} \quad (0 \leq j \leq n) \qquad \cdots(2)$$

The j-th column is replaced by

$$\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$$

[0012] When this Cramer's formula is used, j-th columns of nxn determinant are replaced in order as indicated by the above-mentioned equation (2), thereby making it possible to obtain solutions of the simultaneous linear equations of the equation (1). Namely, if the determinant can be calculated, it is possible to solve the simultaneous linear equations.

[0013] In general, the nxn determinant is expanded as indicated by the equation (3) by using small determinant having degree lower than n. In this case, Δij is expression in which sign given by $(-1)^{i+j}$ is attached to the representation obtained by removing, from the nxn determinant, the i-th row and the j-th column thereof.

$$\begin{vmatrix} a_{00} \cdots\cdots & a_{0n} \\ \vdots & \vdots \\ a_{n0} \cdots\cdots & a_{nn} \end{vmatrix} = a_{0j}\Delta_{0j} + a_{1j}\Delta_{1j} + \cdots\cdots + a_{nj}\Delta_{nj}$$

Removal of the j-th column ...(3)

$$\Delta ij = (-1)i + j \begin{vmatrix} a_{00} \cdots \vdots \cdots & a_{0n} \\ \cdots\cdots\vdots\cdots\cdots & \\ a_{n0} \cdots \vdots \cdots & a_{nn} \end{vmatrix}$$ Removal of the i-th row

[0014] Namely, if small determinants having lower degree are calculated in order, the original determinant can be calculated. Accordingly, if 2x2 determinant which is the determinant of the lowest degree can be calculated, determinant of arbitrary degree can be similarly calculated. In order to calculate 2x2 determinant, it is sufficient to use expansion indicated by the equation (4).

$$\begin{vmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{vmatrix} = a_{00} * a_{11} - a_{01} * a_{10}$$

BNSDOCID: <EP___0930564A1_I_>

[0015] Moreover, in the case of calculating the determinant of 3x3 matrix, expansion indicated by the equation (3) is rewritten into the equation (5).

$$\begin{vmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{vmatrix} = a_{00} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} - a_{10} \begin{vmatrix} a_{01} & a_{02} \\ a_{21} & a_{22} \end{vmatrix} + a_{20} \begin{vmatrix} a_{01} & a_{02} \\ a_{11} & a_{12} \end{vmatrix} \quad \cdots (5)$$

[0016] FIG. 6 shows the state where respective row vectors (a00, a01, a02), (a10, a11, a12), (a20, a21, a22) of 3x3 matrix are stored into registers A0, A1, A2 respectively as 64 bits. The procedure for calculating small determinant of 2x2 by using conventional MM instructions with respect to row vectors stored in this way will be described below.

[0017] FIG. 7 shows the procedure for calculating small determinant of 2x2 by using conventional MM instruction with respect to row vectors of 3x3 matrix of FIG. 6.

[0018] Initially, row vector stored in register A1 is shifted by 16 bits to the right by instruction "SRL B, A1, 16", and is stored into register B.

[0019] Then, product (AND) of the above-mentioned row vector stored in the register B and 000000000000ffff is generated by instruction "ANDIB, Ox000000000000ffff", and is stored into the register B for a second time. Thus, only a11 is stored into 16 bits of low order of bit 0 to bit 15 of the register B.

[0020] Then, row vector stored in the register A1 is shifted by 16 bits to the left by instruction "SLL C, A1, 16", and is stored into register C.

[0021] Then, product (AND) of the above-mentioned row vector stored in the register C and 00000000ffff0000 is generated by instruction "ANDI C, 0x00000000ffff0000", and is stored into the register C for a second time. Thus, only a12 is stored into 16 bits of bit 16 to bit 31 of the register C.

[0022] Then, sum (OR) of data stored in the register B and data stored in the register C is generated by instruction "OR D, B, C", and is stored into register D. Thus, a12, a11 are stored into 32 bits of low order of the register D.

[0023] Then, row vector stored in the register A0 and data stored in the register D are multiplied in parallel by instruction "PMUL E, A0, D", and its result is stored into register E. Namely, a0 1*a12 is stored into 32 bits of high order of the register E, and a02*a11 is stored into 32 bits of low order thereof.

[0024] Then, data stored in the register E is shifted by 32 bits to the right by instruction "SRL F, E, 32", and is stored into register F. Namely, only a01*a12 is stored into 32 bits of low order of the register F.

[0025] Then, product (AND) of the above-mentioned data stored in the register E and 00000000ffffffff is generated by instruction "ANDI E, 0x00000000ffffffff", and is stored into the register E for a second time. Thus, only a02*a11 is stored into 32 bits of low order of the register E.

[0026] Then, by instruction "SUB G, F, E", data stored in the register E is subtracted from data stored in the register F so that its difference is generated. The difference thus obtained is stored into register G. Thus, determinant a01*a12 - a02*a11 of 2x2 matrix is stored into 32 bits of low order of the register G.

[0027] As stated above, in the case where conventional MM instructions are used to calculate determinant of2x2 matrix, the above-mentioned 9 (nine) steps were required.

[0028] Explanation will be given in connection with the case where normal (vector) of triangle is determined as a second more practical example in which the conventional MM instructions are used to carry out parallel operation.

[0029] Three points of the three-dimensional space determine one triangle. Moreover, area of triangle and normal vector are given by absolute value of outer product vector and normalization vector. Outer product of such two three-dimensional vectors is three-dimensional vector given by the equation (6).

CID: <EP___0930564A1_I_>

$$(a_{00}\ a_{01}\ a_{02}) \times (a_{10}\ a_{11}\ a_{12}) = \left(\begin{vmatrix} a_{01} & a_{02} \\ a_{11} & a_{12} \end{vmatrix} \begin{vmatrix} a_{02} & a_{00} \\ a_{12} & a_{10} \end{vmatrix} \begin{vmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{vmatrix} \right) \quad \cdots (6)$$

[0030] FIG. 8 shows the state where two three-dimensional vectors (a00, a01, a02), (a10, a11, a12) are stored in registers A0, A1 as two words respectively consisting of 64 bits. Explanation will be given below in connection with the procedure for calculating outer product by using the conventional MM instructions with respect to two three-dimensional vectors stored in this way.

[0031] FIG. 9 shows the procedure for calculating outer product by using the conventional MM instructions with respect to two three-dimensional vectors of FIG. 8.

[0032] Initially, row vector stored in the register AO is shifted by 16 bits to the right by instruction "SRL B, AO, 16", and is stored into register B.

[0033] Then, row vector stored in the register A0 is shifted by 32 bits to the left by instruction "SLL C, AO, 32", and is stored into register C.

[0034] Then, sum (OR) of data stored in the register B and data stored in the register C is generated by instruction "OR D, B, C", and is stored into register D. Thus, a01, a02, a00, a01 respectively consisting of 16 bits are stored into the register D.

[0035] Then, row vector stored in the register A1 is shifted by 16 bits to the left by instruction "SLL E, A1, 16", and is stored into register E.

[0036] Then, row vector stored in the register A1 is shifted by 32 bits to the right by instruction "SRL F, A1, 32", and is stored into the register F.

[0037] Then, sum (OR) of data stored in the register E and data stored in the register F is generated by instruction "OR G, E, F", and is stored into register G. Thus, a10, a11, a12, a10 respectively consisting of 16 bits are stored into the register G.

[0038] Then, data stored in the register D and data stored in the register G are multiplied in parallel by instruction "PMUL H, D, G", and its result is stored into register H. Namely, a01*a10, a02*a11, a00*a12, a01*a10 respectively consisting of 32 bits are stored into register H.

[0039] Then, row vector stored in the register A0 is shifted by 16 bits to the left by instruction "SLL B, A0, 16", and is stored into register B.

[0040] Then, row vector stored in the register A0 is shifted by 32 bits to the right by instruction "SRL C, A0, 32", and is stored into register C.

[0041] Then, sum (OR) of data stored in the register B and data stored in the register C is generated by instruction "OR D, B, C", and is stored into register D. Thus, a00, a01, a02, a00 respectively consisting of 16 bits are stored into the register D.

[0042] Then, row vector stored in the register A1 is shifted by 16 bits to the right by instruction "SRL E, A1, 16", and is stored into register E.

[0043] Then, row vector stored in the register A1 is shifted by 32 bits to the left by instruction "SLL F, A1, 32", and is stored into register F.

[0044] Then, sum (OR) of data stored in the register E and data stored in the register F is generated by instruction "OR G, E, F", and is stored into register G. Thus, a11, a12, a10, a11 respectively consisting of 16 bits are stored into the register G.

[0045] Then, data stored in the register D and data stored in the register G are multiplied in parallel by instruction "PMUL J, D, G", and its result is stored into register J. Namely, a00*a11, a01*a12, a02*a10, a00*a11 respectively consisting of 32 bits are stored into the register J.

[0046] Then, data stored in the register H is subtracted in parallel from data stored in the register J by instruction "PSUB K, J, H", and its result is stored into register K. Namely, a00*a11 - a01*a10, a01*a12 - a02*a11, a02*a10 - a00*a12, a00*a11 - a10*a10 respectively consisting of 32 bits are stored into the register K.

[0047] As stated above, in the case where the conventional MM instructions are used to calculate outer product of two three-dimensional vectors, the above-mentioned 15 steps were required.

[0048] Explanation will now be given in connection with the case of calculating inner product of two vectors as a third more practical example in which the conventional MM instructions are used to carry out parallel operation.

[0049] Inner product of two vectors represents degree of correlation therebetween. As such inner product of two vec-

BNSDOCID: <EP___0930564A1_I_>

tors, inner product of, e.g., two four-dimensional vectors is given by the equation (7).

$$(a_0a_1a_2a_3)*(b_0b_1b_2b_3) = a_0*b_0 + a_1*b_1 + a_2*b_2 + a_3*b_3 \quad (7)$$

[0050] FIG. 10 shows the state where two four-dimensional vectors (a0, a1, a2, a3), (b0, b1, b2, b3) of 64 bit word are respectively stored into registers A, B as two words. Explanation will be given below in connection with the procedure for calculating inner product by using conventional MM instructions with respect to two four-dimensional vectors stored in this way.

[0051] FIG. 11 shows procedure for calculating inner product by using conventional MM instructions with respect to two four-dimensional vectors of FIG. 10. In this example, the portions labeled mark x indicate that values irrelevant to this operation are stored.

[0052] Initially, data stored in register A and data stored in register B are multiplied in parallel by instruction "PMUL C, A, B", and its result is stored into register C. Namely, a0*b0, a1*b1, a2*b2, a3*b3 respectively consisting of 16 bits are stored into the register C.

[0053] Then, data stored in the register C is shifted by 16 bits to the left by instruction "SLL D, C, 16", and is stored into register D.

[0054] Then, data stored in the register C and data stored in the register D are added in parallel by instruction "PADD E, C, D", and its result is stored into register E. Thus, within the register E, a2*b2 + a3*b3 of 16 bits are stored into bit 16 to bit 31, and a0*b0 + a1*b1 of 16 bits are stored into bit 48 to bit 63.

[0055] Then, data stored in the register E is shifted by 32 bits to the left by instruction "SLL F, E, 32", and is stored into register F. Thus, within the register F, only a2*b2 + a3*b3 are stored into the Most Significant 16 bits, and two data values of 16 bits of low order both become equal to zero.

[0056] Then, data stored in the register E and data stored in the register F are added in parallel by instruction "PADD G, E, F", and its result is stored into register G. Thus, within the register G, a0*b0 + a1*b1 + a2*b2 + a3*b3 are stored into the Most Significant 16 bits, and $a_2*b_2 + a_3*b_3$ are stored into bit 16 to bit 31.

[0057] As stated above, in the case where the conventional MM instructions are used to calculate inner product of two four-dimensional vectors, the above-mentioned 5 steps were required.

[0058] Meanwhile, in the arithmetic apparatus and the arithmetic method using conventional MM instructions, while data of plural fields of n bits are stored in the register, operation is performed only between the same (corresponding) bit fields of these fields. Namely, since arithmetic operation cannot be directly implemented between fields within word subject to operation consisting of plural fields, there took place necessity of carrying out extra field operation for performing operation between desired fields in carrying out parallel operation as described above, thus failing to cause the operation speed to be sufficiently high.

Disclosure of the Invention

[0059] This invention has been made in view of the above-described problems and its object is to provide an arithmetic apparatus and an arithmetic method which can perform parallel operation at a high speed with the number of steps lesser than that of the conventional arithmetic apparatus.

[0060] The arithmetic apparatus according to this invention comprises arithmetic and logic means for performing arithmetic and logical operation with respect to word subject to operation constituted with plural fields consisting of M bits ($M \geq 1$), shift processing means for implementing shift operation by a predetermined number of bits with respect to the word subject to operation, and a register for storing the word subject to operation and word in which the operation has been carried out, and has a function to perform parallel operation between the plural fields within the same word subject t operation.

[0061] In addition, an arithmetic method according to this invention is directed to an arithmetic method for performing arithmetic and logical operation in field units with respect to word subject to operation constituted with plural fields consisting of M bits, wherein the method includes a step of exchanging two fields or more within the same word subject to operation.

[0062] In accordance with the arithmetic apparatus and the arithmetic method as stated above, since there is no necessity of carrying out extra field operation, it is possible to perform parallel operation at a high speed by the number of steps lesser than that of the prior art.

Brief Description of the Drawings

[0063]

FIG. 1 is a view showing an example of the configuration of conventional CPU.

BNSDOCID: <EP___0930564A1_I_>

FIG. 2 is a view for explaining multiplication by multiplier of 64 bits x 64 bits.

FIG. 3 is a view for explaining parallel multiplication by quartered multiplier of 64 bits x 64 bits.

FIG. 4 is a view for explaining addition by adder of 64 bits x 64 bits.

FIG. 5 is a view for explaining parallel addition by quartered adder of 64 bits x 64 bits.

FIG. 6 is a view showing the state where row vectors of 3x3 matrix are stored in registers respectively as 64 bit words.

FIG. 7 is a view showing the procedure for calculating small determinant of 2x2 by using conventional MM instructions with respect to row vectors of 3x3 matrix.

FIG. 8 is a view showing the state where two three-dimensional vectors are stored in registers respectively as words of 64 bits.

FIG. 9 is a view showing the procedure for calculating outer product by using conventional MM instructions with respect to two three-dimensional vectors.

FIG. 10 is a view showing the state where two four-dimensional vectors are stored in registers respectively as two words.

FIG. 11 is a view showing the procedure for calculating inner product by using conventional MM instructions with respect to two four-dimensional vectors.

FIG. 12 is a view showing an example of the configuration of CPU which is one form of an arithmetic apparatus of this invention.

FIG. 13 is a view showing an example of fundamental configuration of CPU having MM instructions.

FIGS. 14A, B, C are views for explaining instructions "PMUL" and "PADD".

FIGS. 15A to E are views for explaining MM instructions of the arithmetic apparatus of this invention.

FIG. 16 is a view showing an example of the configuration of data exchange unit (EXC circuit).

FIG. 17 is a view for explaining multiplexer (MUX) of the EXC circuit.

FIG. 18 is a view showing two commands sent to the MUX and the operation thereof.

FIG. 19 is a view showing correspondence relationship between EXC command sent to the EXC circuit and MM instruction to be realized.

FIG. 20 is a view showing a circuit for realizing instruction "PEXC".

FIG. 21 is a view showing a circuit for realizing instruction "PEXH".

FIG. 22 is a view showing a circuit for realizing instruction "PROT3".

FIG. 23 is a view showing a circuit for realizing instruction "PHADD".

FIG. 24 is a view showing a circuit for realizing instruction "PHSUB".

FIG. 25 is a view showing procedure for calculating, by the arithmetic apparatus of this invention, small determinant of 2x2 with respect to row vectors of 3x3 matrix respectively stored.

FIG. 26 is a view showing procedure for calculating outer product with respect to two three-dimensional vectors by

BNSDOCID: <EP___0930564A1_I_>

the arithmetic apparatus of this invention.

FIG. 27 is a view showing procedure for calculating inner product with respect to two four-dimensional vectors by the arithmetic apparatus of this invention.

FIG. 28 is a block diagram showing an example of the configuration of a picture preparation apparatus to which the arithmetic unit (apparatus) according to this invention is applied.

Best Mode for Carrying Out the Invention

[0064] Preferred embodiments of an arithmetic apparatus and an arithmetic method of this invention will be described below with reference to the attached drawings. In the following description, the configuration of the embodiment of the arithmetic apparatus of this invention will be first explained and the embodiment of the arithmetic method of this invention will be explained with reference to that configuration.

[0065] FIG. 12 shows an example of the configuration of the main part of CPU as one embodiment of the arithmetic apparatus of this invention. This CPU is of the configuration comprising an arithmetic and logic unit (ALU) 330 serving as arithmetic and logic means, a shift processing unit (SHT) 340, and a register unit (REG) 350, wherein these units can mutually transfer data through buses (BUS) 360, 370, 380 of 64 bits and parallel buses of 16 bits. The above-mentioned ALU 330, the SHT 340 and the REG 350 are of the configuration in which they are respectively divided into four sections.

[0066] While the above-mentioned respective components have configuration similar to the respective portions of the CPU shown in FIG. 13, the former differs from the latter in that the former comprises data exchange units (EXC) 310, 320 serving as bit field exchange means within word. Namely, by the EXCs 310, 320 serving as bit field exchange means within this word, there is realized arithmetic (computational) function to perform operation between plural fields within the same word subject to operation at the ALU 330. In this example, one field consists of M bits ($M \geq 1$). In the embodiment described below, one field is caused to be, e.g., 16 bits.

[0067] Prior to explanation with respect to new MM instructions that the above-described arithmetic apparatus of this invention has, "PMUL" and "PADD" which are previously described MM instructions will be described for a second time with reference to an example of the configuration of the Central Processing unit (CPU) which is essential point of the arithmetic unit (apparatus) of this invention.

[0068] FIG. 13 shows an example of the fundamental configuration of CPU having MM instructions. This example of the configuration of the CPU having MM instructions is based on the example of the configuration of the conventional CPU which has not new MM instructions shown in FIG. 1, but differs from the latter in that an ALU 230, a SHT 240 and a REG 250 are respectively divided into four sections.

[0069] Further, as data transfer path between a bus 260 and the ALU 230, four 16 bit parallel transfer paths 265 are provided in place of parallel transfer paths of 64 bits.

[0070] FIG. 14A to C show MM instructions "PMUL" and "PADD" executed at the arithmetic unit of FIG. 13.

[0071] FIG. 14A shows the state where data of respective 16 bits are respectively stored into quartered respective fields of 16 bits of 64 bit registers A, B of the REG 250.

[0072] FIG. 14B shows the state where four data individually stored in four fields of the register A and four data stored in the register B are multiplied in parallel at the ALU 230 by instruction "PMUL C, A, B", and products respectively consisting of 32 bits are stored into register C of the REG 250.

[0073] Moreover, FIG. 14C shows the state where four data stored in the register A and four data stored in the register B are added in parallel by instruction "PADD C, A, B" and sums respectively consisting of 16 bits are stored into register C.

[0074] However, operations by MM instructions as described above in the arithmetic unit of FIG. 13 are carried out in word units, and the number of steps was additionally required for the purpose of performing operation in field units. In view of the above, the arithmetic unit of this invention is caused to be of the configuration further including "bit field exchange instruction within the word" and "operation instruction between data within the word" which are new MM instructions and adapted for performing operation by lesser number of steps.

[0075] The group of MM instructions of the arithmetic unit (apparatus) of this invention will now be described with reference to FIG. 15A to E.

[0076] FIG. 15A shows instruction "PEXC". Namely, the instruction "PEXC 19, A" serves to exchange, in the state where data of the Most Significant field and data of the Least Significant field of quartered register A are caused to remain as they are, data of two fields at the central portion therebetween to store them into register B.

[0077] FIG. 15B shows instruction "PEXH". Namely, the instruction "PEXH B, A" serves to exchange each other respective data of two fields of high order of the quartered register A, and to exchange each other respective data of two fields of low order thereof to store them into the register B.

BNSDOCID: <EP___0930564A1_I_>

[0078] FIG. 15C shows instruction "PROT3". Namely, the instruction "PROT3 B, A, 16" serves to allow data of the Most Significant field of the quartered register A to remain as it is and to shift, by 16 bits, respective data of three fields of low order to allow such data to undergo rotation to store them into the register B.

[0079] FIG. 15D shows instruction "PHADD". Namely, the instruction "PHADD B, A" serves to add each other respective data of two fields of high order of the quartered register A, and to add each other respective data of two fields of low order thereof to store them into the register B.

[0080] FIG. 15E shows instruction "PHSUB". Namely, the instruction "PHSUB B, A" serves to allow respective data of two fields of high order of the quartered register A to undergo subtractive processing, and to allow respective data of two fields of low order to undergo subtractive processing to store them into the register B.

[0081] As stated above, the arithmetic unit (apparatus) of this invention further has, in addition to the conventional MM instructions, instruction for carrying out exchange between divided bit fields and instruction for performing operation between different bit fields within the same register to thereby improve operation performance.

[0082] The configuration of the arithmetic unit (apparatus) of this invention further having new MM instructions as described above in addition to the conventional MM instructions will now be described in more practical sense.

[0083] FIG. 16 shows an example of the configuration of data exchange unit (EXC circuit) 310 of FIG. 12. Respective inputs A0 to A3 to this EXC circuit 310 are delivered to respective multiplexers (MUXs) 311 to 314. Further, the respective MUXs select data to be outputted by respective two commands delivered thereto. Thus, the operation of the EXC 310 is controlled by commands C0 to C7.

[0084] It is to be noted that while only the EXC 310 has been described here, the operation similarly applies to the EXC circuit 320.

[0085] The MUXs 311 to 314 of the above-described EXC circuits 310, 320 will now be described. These MUXs have configuration of four inputs and one output, and their operations are controlled by respective two commands.

[0086] FIG. 17 shows MUX 311 among the above-mentioned MUXs 311 to 314. This MUX 311 has configuration of four inputs and one output, and its operation is controlled by two commands C0, C1.

[0087] FIG. 18 shows correspondence relationship between two commands sent to the MUX 311 and the operation. Namely, when commands C0, C1 are both 0, input A0 is caused to be output B0. Moreover, when C0 is 0 and C1 is 1, input A1 is caused to be B0. Similarly, when C0 is 1 and C1 is 0, input A2 is caused to be output B0. In addition, when C0, C1 are both 1, input A3 is caused to be output B0.

[0088] It is to be noted that while the MUX 311 has been described here, the operation similarly applies to MUXs 312 to 314. Namely, the operation of the MUX 312 is controlled by commands C2, C3, the operation of the MUX 313 is controlled by commands C4, C5, and the operation of the MUX 314 is controlled by commands C6, C7.

[0089] FIG. 19 shows the correspondence relationship between EXC commands C0 to C7 sent to the EXC circuit shown in FIG. 16 and MM instructions realized by these commands. Namely,

When C0, C1, C3 and C4 are 0, and C2, C5, C6 and C7 are 1, instruction "PEXC" is realized.

When C0, C2, C3 and C7 are 0, and C1, C4, C5 and C6 are 1, instruction "PEXH" is realized.

When C0, C1, C4 and C7 are 0, and C2, C3, C5 and C6 are 1, instruction "PROT3" is realized.

When C0, C2, C3 and C7 are 0, and C1, C4, C5 and C6 are 1, instruction "PHADD" is realized.

When C0, C2, C3 and C7 are 0, and C1, C4, C5 and C6 are 1, instruction "PHSUB" is realized.

[0090] It is to be noted that the above-mentioned instructions "PHADD" and "PHSUB" are the same with respect to the EXC instruction, but are different in command of ALU.

[0091] Explanation will now be given in more practical sense in connection with a circuit for realizing new MM instructions that the above-described arithmetic unit (apparatus) of this invention has. In the following description, a0 to a3 are input data respectively having data width of 16 bits or 32 bits, and constitute one word as a whole. In addition, b0 to b3 are output data respectively having data width of 16 bits or 32 bits, and constitute one word as a whole.

[0092] FIG. 20 shows a circuit for realizing instruction "PEXC". This circuit is of the configuration comprising exchange circuit "exchange". With respect to four data a0, a1, a2, a3 inputted to this circuit, the Most Significant data a0 and the Least Significant data a3 are respectively outputted as b0 and b3 as they are. In addition, two data between the Most Significant data and the Least Significant data are exchanged each other, and they are outputted in the state where a1 is caused to be b2 and a2 is caused to be b1.

[0093] FIG. 21 shows a circuit for realizing instruction "PEXH". This circuit is of the configuration comprising two exchange circuits "exchange". Two data a0, a1 of high order of four data a0, a1, a2, a3 inputted to this circuit are exchanged each other, and they are outputted in the state where a0 is caused to be b1 and a1 is caused to be b0. In

BNSDOCID: <EP___0930564A1_I_>

addition, two data a2, a3 of low order of the above-mentioned inputted four data are exchanged each other, and they are outputted in the state where a2 is caused to be b3 and a3 is caused to be b2.

[0094] FIG. 22 shows a circuit for realizing instruction "PROT3". In this example, "SELECT" is a selector circuit. The Most Significant data a0 of four data a0, a1, a2, a3 inputted to this circuit is outputted in the state caused to be b0 as it is. In addition, other three data a1, a2, a3 are outputted in the state where, e.g., a1 is caused to be b3, a2 is caused to be b1 and a3 is caused to be b2 by the selector circuit "select" of three inputs and one output. Namely, the above-mentioned three data except for the Most Significant data a0 are outputted after undergone rotation.

[0095] FIG. 23 shows a circuit for realizing instruction "PHADD". This circuit is of the configuration comprising two adding circuits "ADD". Two data a0, a1 of high order of four data a0, a1, a2, a3 inputted to this circuit are added to each other, and are outputted in the state caused to be b0. In addition, two data a2, a3 of low order of the above-mentioned inputted four data are exchanged each other, and are outputted in the state caused to be b2.

[0096] FIG. 24 shows a circuit for realizing instruction "PHSUB". This circuit is of the configuration comprising two subtracting circuits "SUB". Two data a0 and a1 of high order of four data a0, a1, a2, a3 inputted to this circuit are outputted in the state where data a1 is subtracted from data a0 so that its difference is caused to be b0. In addition, two data a2 and a3 of low order of the inputted four data are outputted in the state where data a3 is subtracted from data a2 so that its difference is caused to be b2.

[0097] Explanation will now be given in connection with the case where operation is performed by the arithmetic unit (apparatus) of this invention having a function to carry out exchange an/or operation (computation) of different bit fields within the same word as previously described.

[0098] FIG. 25 shows the procedure for calculating small determinant of 2x2 with respect to row vectors of 3x3 matrix by using the arithmetic unit (apparatus) of this invention.

[0099] Initially, by the previously described instruction "PEXH D, A1", two data of high order of the quartered register A1 are exchanged each other, and two data of low order thereof are exchanged each other thus to store them into register D.

[0100] Then, parallel multiplication of row vector stored in register A0 and data stored in the register D is carried out in 16 bit units by instruction "PMULH E, A0, D", and its result is stored into register E. This instruction "PMULH" is instruction for carrying out operation similar to the previously described instruction "PMUL" with only half of the word length being as unit. Thus, a01*a12 is stored into 32 bits of high order of the register E and a02*a11 is stored into 32 bits of low order thereof

[0101] Then, parallel subtraction to subtract, from data of high order stored in the register E, data of low order stored in the register E is carried out, in 32 bit units, by instruction "PSUBW G, E", and its result is stored into register G. This instruction "PSUBW" is instruction for carrying out operation similar to operation "PSUB" with word length being as unit. Thus, 0 is stored into 32 bits of high order of the register G and a01*a12 - a02*a11 is stored into 32 bits of low order.

[0102] As stated above, in order to calculate the 2x2 determinant, with the conventional arithmetic unit (apparatus), 9 steps were required as shown in FIG. 7. On the contrary, in accordance with the arithmetic unit (apparatus) of this invention, such calculation can be performed only by the above-mentioned three steps.

[0103] FIG. 26 shows the procedure for calculating outer product of two three-dimensional vectors by the arithmetic unit (apparatus) of this invention.

[0104] Initially, by instruction "PROT3 B, A0, 16", the Most Significant data of register A0 is caused to be as it is and three data of low order are shifted by 16 bits to allow those data to undergo rotation to store them into register B.

[0105] Then, by instruction "PROT3 C, A1, 32", the Most Significant data of register A1 is caused to be as it is and three data of low order are shifted only by 32 bits to allow those data to undergo rotation to store them into register C.

[0106] Then, by instruction "PMUL D, B, C", parallel multiplication of row vector stored in the register B and data stored in the register C is carried out. The result thus obtained is stored into register D. Namely, 0 is stored into the Most Significant 32 bits of the register D, and a02*a11, a00*a12 and a01*a10 are stored in order into subsequent (succeeding) respective 32 bits.

[0107] Then, by instruction "PROT3 B, A0, 32", the Most Significant data of the register A0 is caused to be as it is and three data of low order are shifted only by 32 bits to allow those data to undergo rotation to store them into the register B.

[0108] Then, by instruction "PROT3 C, A1, 16", the Most Significant data of the register A1 is caused to be as it is and three data of low order are shifted only by 16 bits to allow those data to undergo rotation to store them into the register C.

[0109] Then, by instruction "PMUL E, B, C", parallel multiplication of data stored in the register B and data stored in the register C is carried out. The result thus obtained is stored into the register E. Namely, 0 is stored into the Most Significant 32 bits of the register E and a01*a12, a02*a10 and a00*a11 are stored in order into subsequent (succeeding) resp ctive 32 bits.

[0110] Then, by instruction "PSUB F, E, D", parallel subtraction to subtract, from data stored in the register E, data stored in the register D is carried out. The result thus obtained is stored into register F. Namely, 0 is stored into the Most

BNSDOCID: <EP___0930564A1_I_>

Significant 32 bits of the register F and a01*a12 - a02*a11, a02*a10 - a00*a12, a00*a11 - a01*a10 are stored into subsequent (succeeding) respective 32 bits.

[0111] As stated above, in order to calculate outer product of two three-dimensional vectors, 15 steps were required as shown in FIG. 9 in the conventional arithmetic apparatus. On the contrary, in accordance with the arithmetic unit (apparatus) of this invention, such calculation can be performed only by the seven steps.

[0112] FIG. 27 shows the procedure for calculating inner product of two four-dimensional vectors by the arithmetic unit (apparatus) of this invention.

[0113] Initially, by instruction "PMUL C, A, B", parallel multiplication of data stored in register A and data stored in register B is carried out. The result thus obtained is stored into register C. Namely, a0*b0, a1*b1, a2*b2, a3*b3 respectively consisting of 32 bits are stored into register C.

[0114] Then, by instruction "PHADD D, C", two data of high order of the register C are added to each other and two data of low order thereof are added to each other to store them into register D.

[0115] Then, instruction "PEXC E, D", the Most Significant data and the Least Significant data of the register D are caused to be as they are, and two data at the central portion therebetween are exchanged to store them into register E.

[0116] Then, by instruction "PHADD G, E", two data of high order of the register E are added to each other and two data of low order thereof are added to each other to store them into register G. Thus, a0*b0 + a1*b1 + a2*b2 + a3*b3 are stored into the

Most Significant 32 bits of the register G.

[0117] In this example, the portions labeled mark x of FIG. 27 indicate that values irrelevant to this operation are stored.

[0118] As stated above, in order to calculate inner product of two four-dimensional vectors, 5 steps were required as shown in FIG. 11 in the conventional arithmetic unit (apparatus). On the contrary, in accordance with the arithmetic unit (apparatus) of this invention, such a calculation can be performed only by the four steps.

[0119] FIG. 28 shows an example of the configuration of a picture preparation apparatus constituted with the arithmetic unit (apparatus) according to this invention having the above explained MM instructions.

[0120] In FIG. 28, a CPU1 which is Central Processing Unit comprised of microprocessor, etc. serves to take out operation information of an input device 4 such as input pad or joy stick etc. through an interface 3 and a main bus 9, and the arithmetic unit of this invention is used for this CPU1. Further, the CPU1 sends, on the basis of the operation information thus taken out, information of three-dimensional picture stored in a main memory 2 which is first memory to a graphic processor 6 through the main bus 9.

[0121] The graphic processor 6 serves to convert sent information of three-dimensional picture to generate picture data, and three-dimensional picture by picture data generated here is depicted on a video memory 5 which is second memory. Three-dimensional picture data depicted on this video memory 5 is read out at the time of scanning of video signal. Thus, three-dimensional picture is displayed on display unit (not shown).

[0122] Moreover, simultaneously with displaying three-dimensional picture as described above, speech (sound) information corresponding to the displayed three-dimensional picture within the operation information which has been taken out by the CPU1 is sent to an audio processor 7. The audio processor 7 displays, on the basis of this sent speech information, speech data stored in an audio memory 8.

[0123] Such a picture preparation apparatus is used, e.g., in home game machines for which it is required to display three-dimensional picture with relatively high accuracy and at high speed.

[0124] In the home game machines, as a method of displaying three-dimensional picture by using a picture preparation apparatus as described above, the shading method of adding shade of object to be displayed and the texture mapping of deforming any other two-dimensional picture to paste it are representative.

[0125] Moreover, there are many instances where, as the coordinate system representing three dimensions, there are used object coordinate system for representing shape or dimension relating to three-dimensional object itself, world coordinate system indicating position of object when three-dimensional object is disposed in space, and screen coordinate system for representing three-dimensional object displayed on screen. There are many instances where particularly, polygonal area serving as unit which represents three-dimensional picture of three-dimensional object on the screen coordinate system, which is so called polygon, is dealt as simplified triangular area.

[0126] The arithmetic unit (apparatus) according to this invention is suitable, with respect to such triangular area (polygon), for calculating vertex coordinates, or carrying out inner product calculation, etc. of normal vector and light source vector from attribute of object and light source data.

[0127] In accordance with arithmetic apparatus as explained above, this apparatus is caused to be of the configuration further having, in addition to the conventional MM instrudtions, MM instructions having a function to perform operation between plural fields within the same word of operational object (object to be computed). For this reason, it is possible to perform parallel operation at a high speed by the number of steps lesser than that of the prior art.

BNSDOCID: <EP___0930564A1_I_>

[0128] It is to be noted that this invention is not limited to the above-described embodiments, but it is a matter of course that, e.g., the number of bits of register and/or the number of bits of field are not limited to the numbers shown.

Claims

1. An arithmetic apparatus comprising;

   arithmetic and logic means for performing arithmetic and logical operation with respect to word subject to operation, which is constituted with plural fields consisting of M bits ($M \geq 1$);
   shift processing means for implementing shift operation by a predetermined number of bits with respect to the word subject to operation; and
   a register for storing the word subject to operation and word in which the operation has been carried out,
   wherein the apparatus has a function to perform parallel operation between the plural fields within the same word subject to operation.

2. An arithmetic apparatus as set forth in claim 1,

   wherein the arithmetic and logic means includes plural arithmetic and logic units for performing arithmetic and logical operation in units of the field with respect to data subject to operation, the shift processing means includes a shift processing unit for implementing shift operation by a predetermined number of bits in the field units with respect to data subject to operation, and the register includes plural register units for storing, in the field units, data subject to operation and data in which operation has been carried out.

3. An arithmetic apparatus as set forth in claim 2,

   which further comprises field exchange means for carrying out exchange between (predetermined ones of) the fields within word subject to operation consisting of the plural fields.

4. An arithmetic method for performing arithmetic and logical operation in field units with respect to word subject to operation constituted with plural fields consisting of M bits ($M \geq 1$),

   the method including a step of exchanging two fields or more within the same word subject to operation.

5. An arithmetic method as set forth in claim 4,

   wherein arithmetic and logical operation is carried out between the fields of word subject to operation in which field exchange has been carried out to store result of the operation into one of the fields subject to operation.

BNSDOCID: <EP___0930564A1_I_>

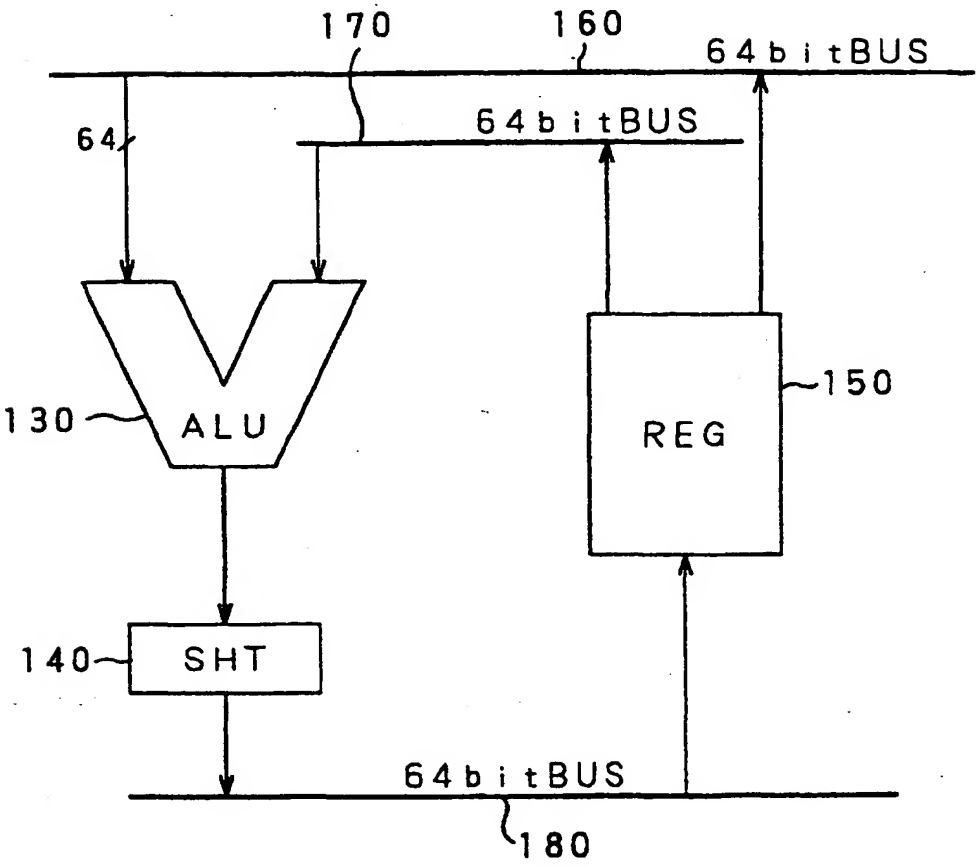

FIG.1

BNSDOCID: <EP___0930564A1_I_>

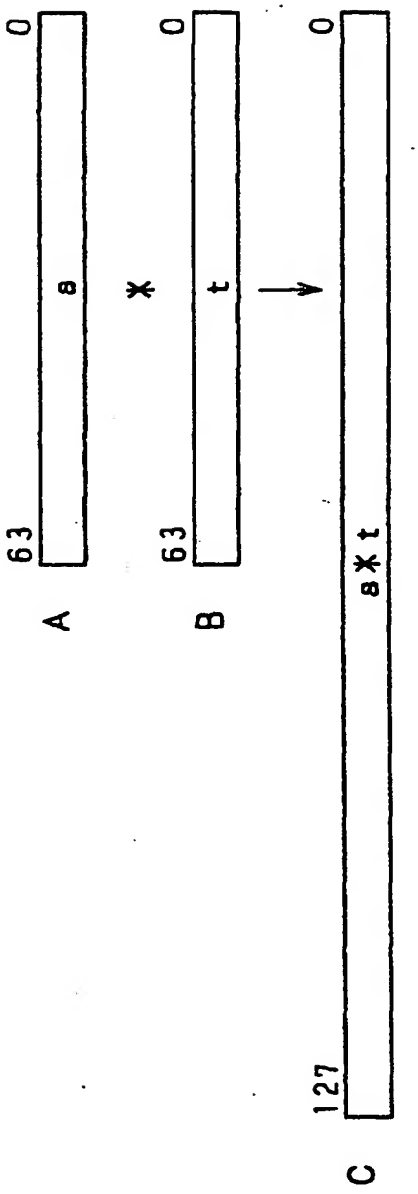

FIG.2

BNSDOCID: <EP___0930564A1_I_>

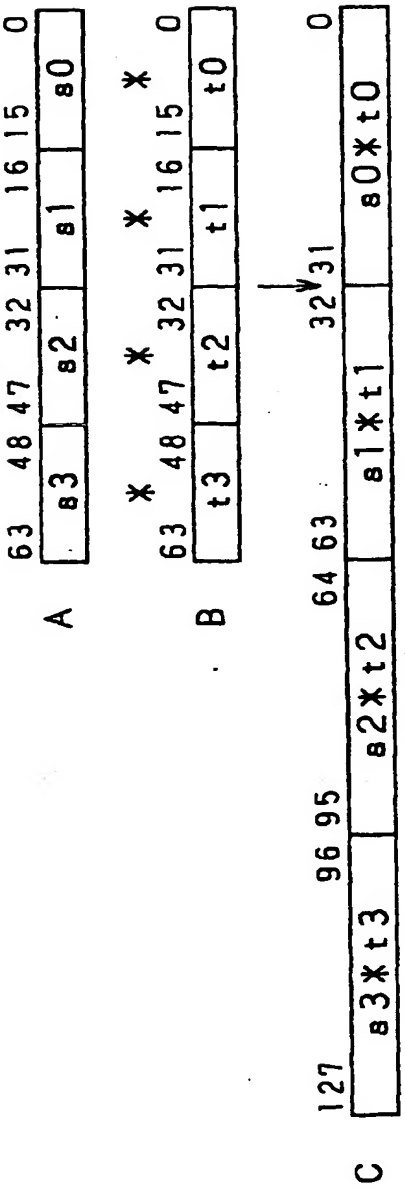

FIG.3

BNSDOCID: <EP___0930564A1_I_>

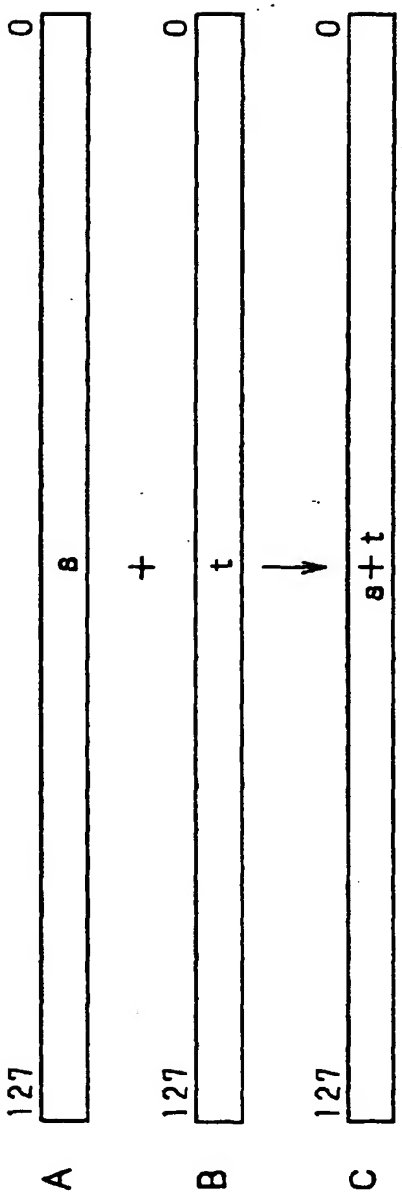

FIG.4

BNSDOCID: <EP___0930564A1_I_>

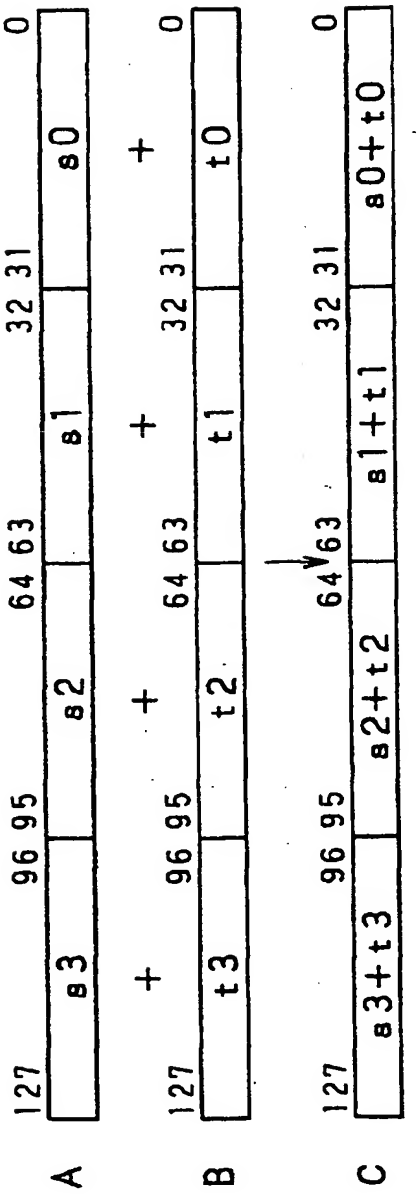

FIG.5

BNSDOCID: <EP___0930564A1_I_>

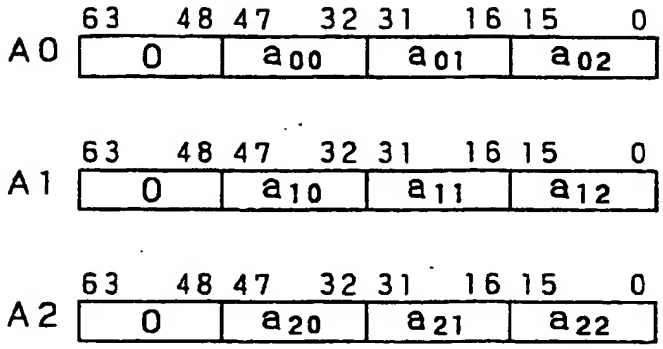

FIG.6

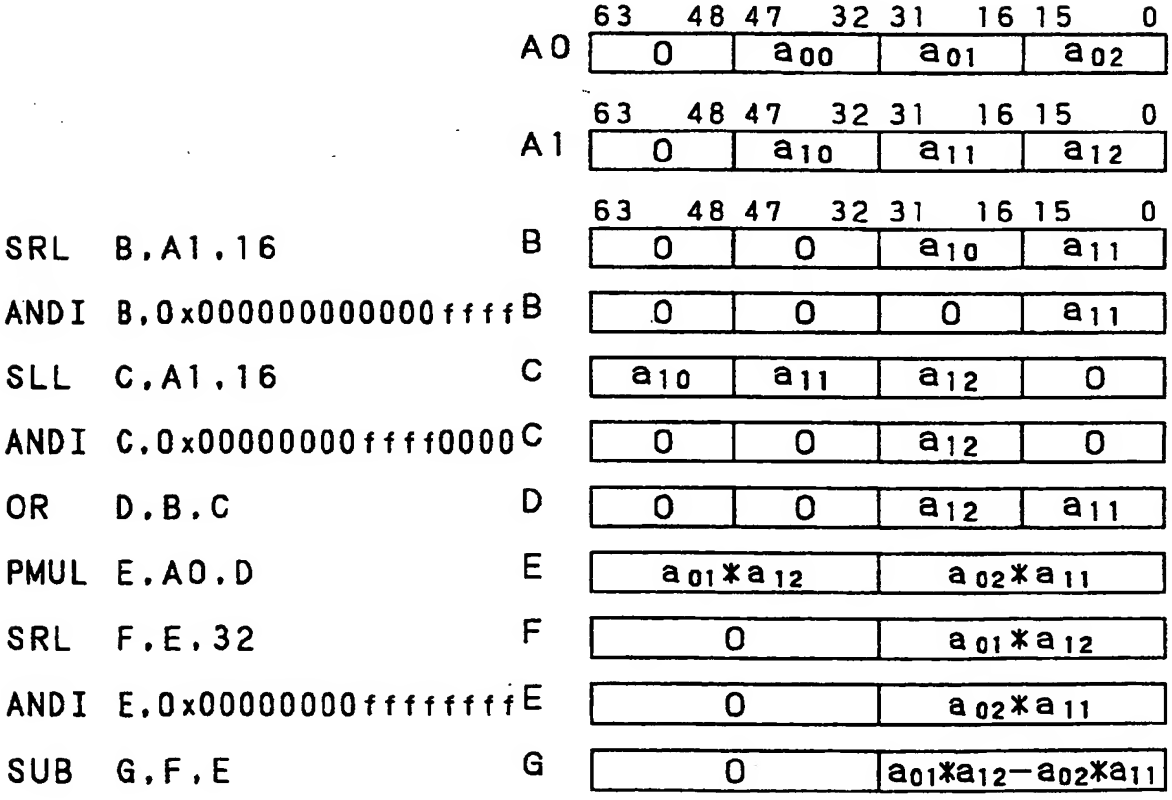

FIG.7

BNSDOCID: <EP___0930564A1_I_>

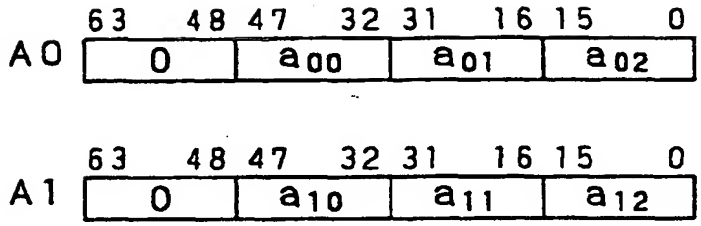

FIG.8

BNSDOCID: <EP___0930564A1_I_>

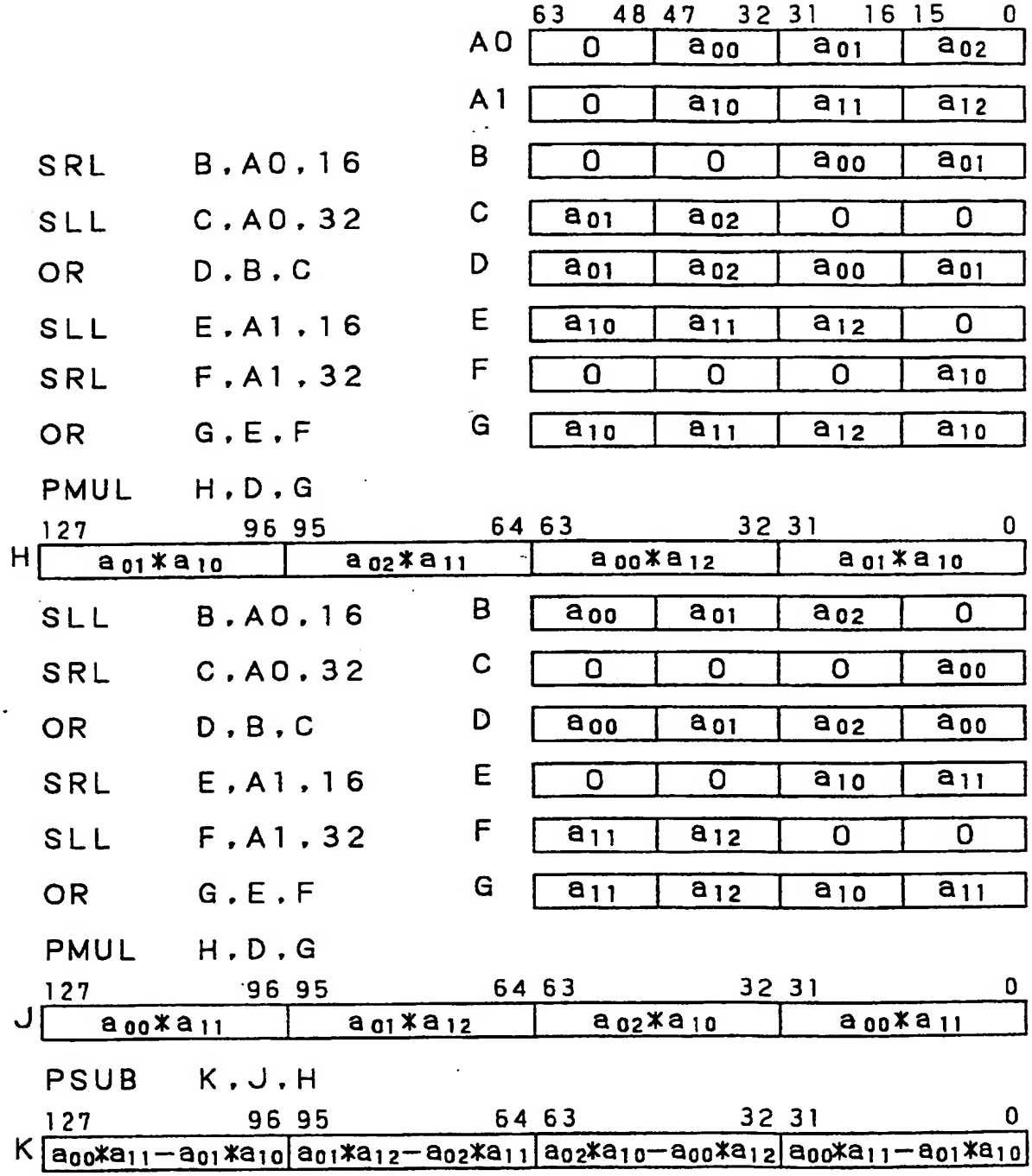

FIG.9

BNSDOCID: <EP___0930564A1_I_>

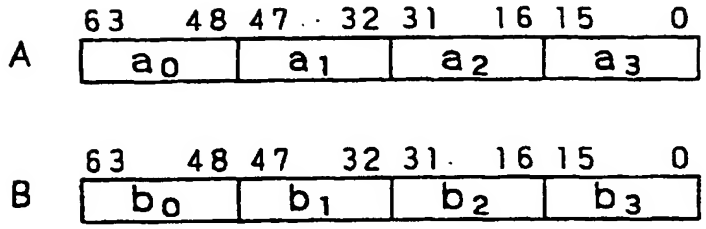

FIG.10

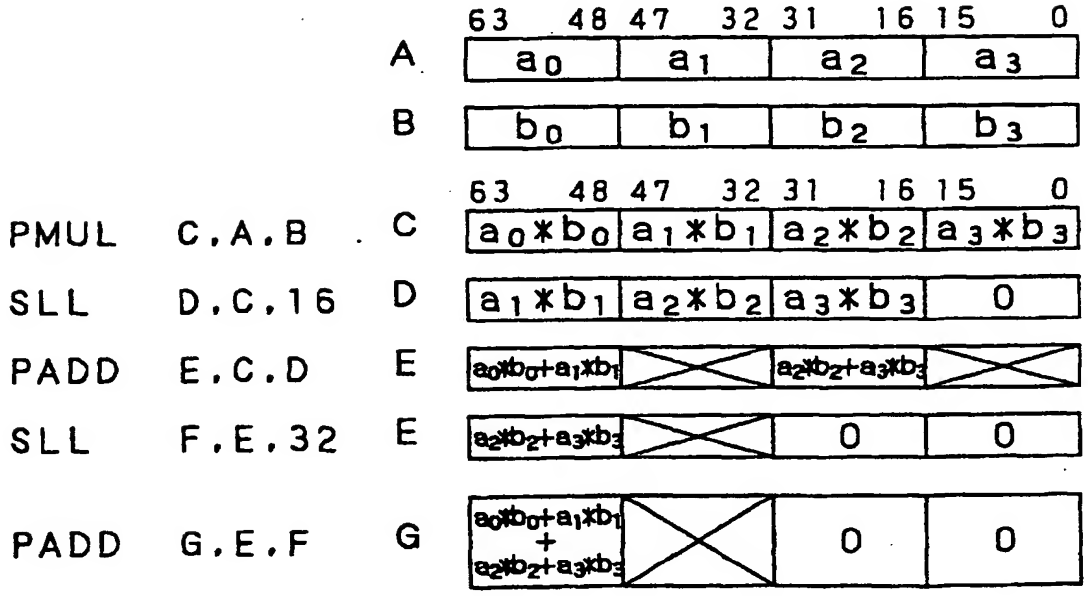

FIG.11

BNSDOCID: <EP___0930564A1_I_>

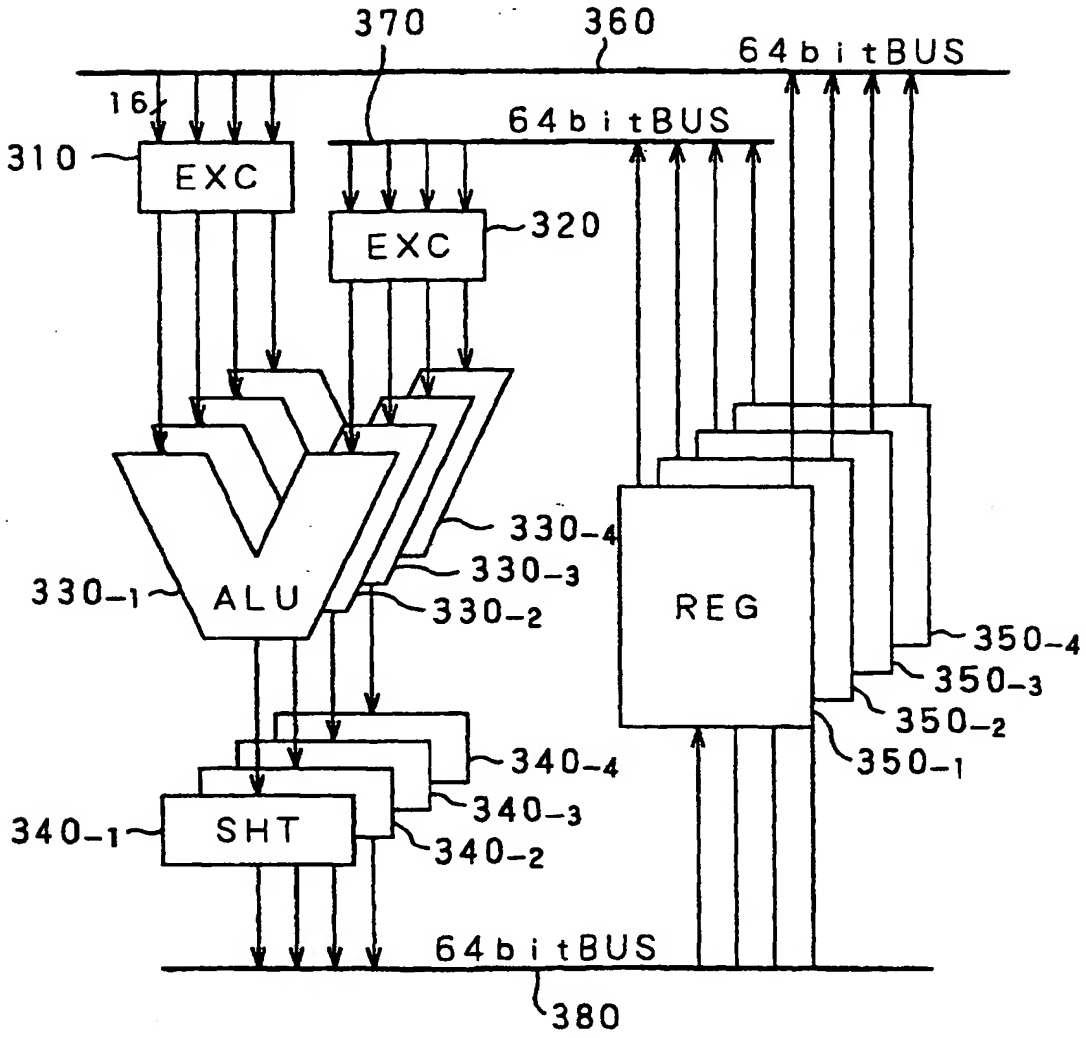

FIG.12

BNSDOCID: <EP___0930564A1_I_>

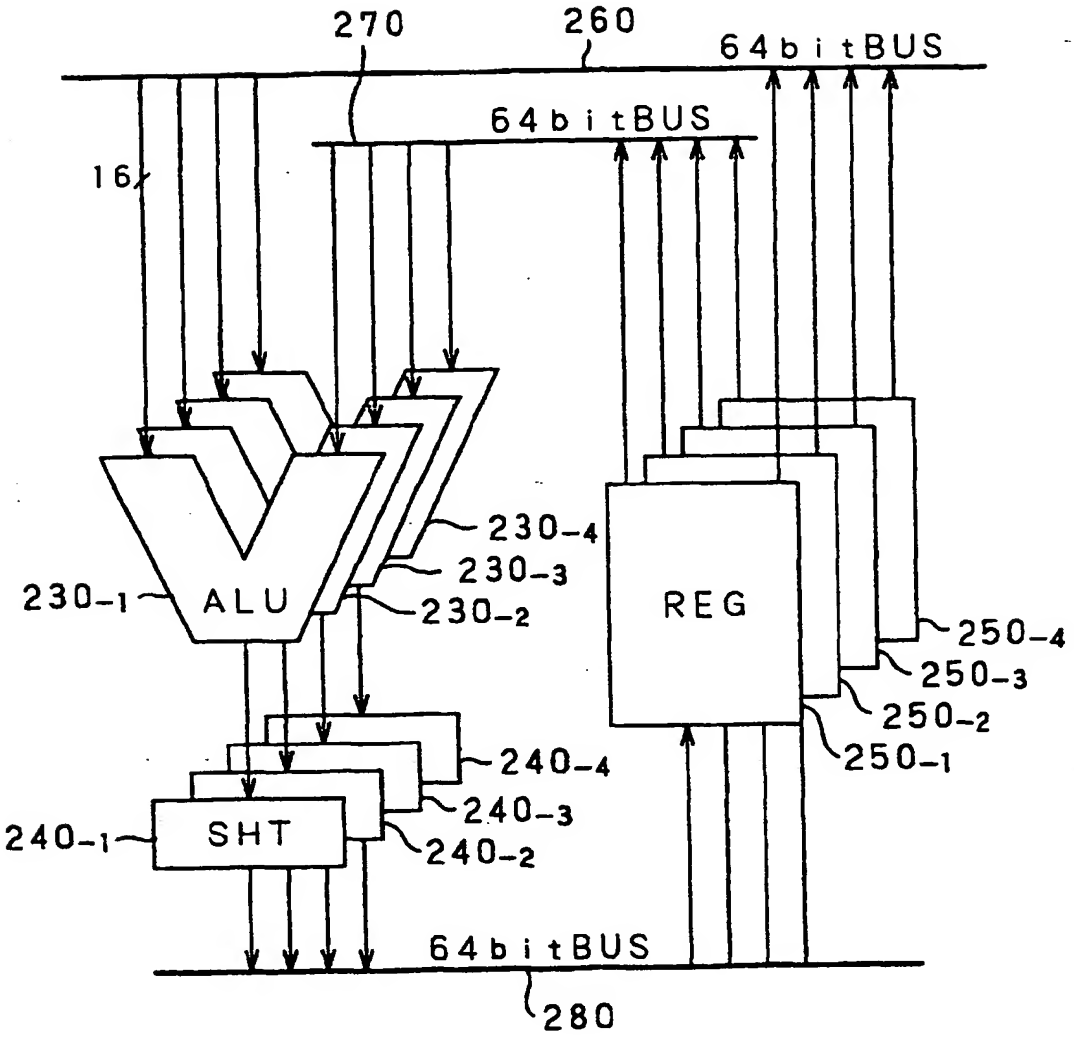

FIG.13

BNSDOCID <EP 0930564A1 >

FIG.14A

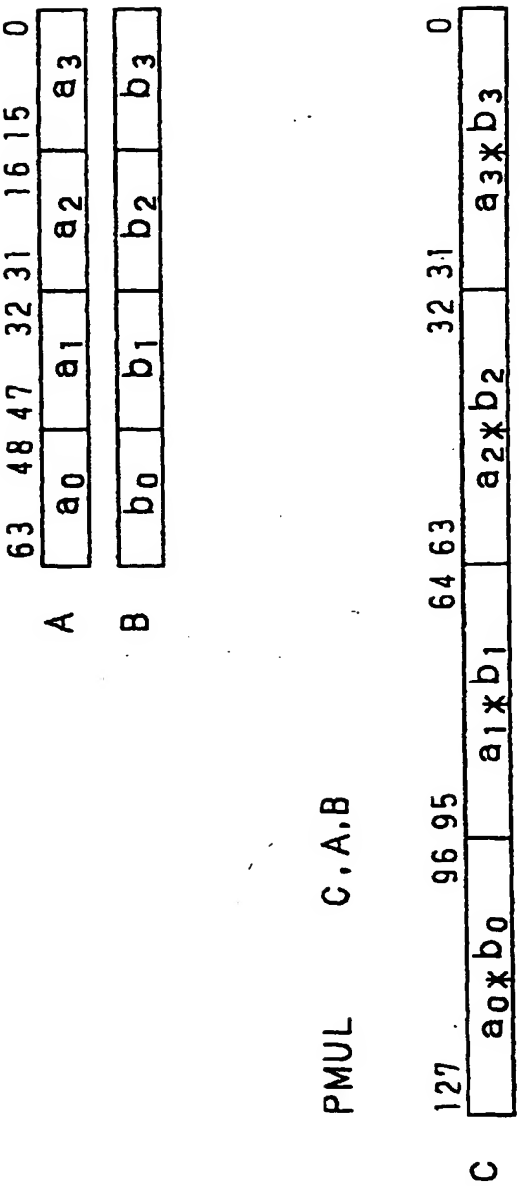

FIG.14B

FIG.14C

BNSDOCID: <EP___0930564A1_I_>

PEXC B.A

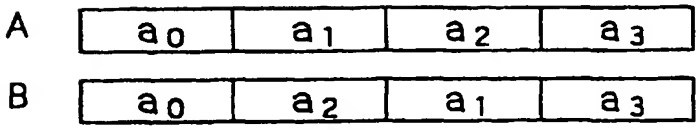

FIG.15A

PEXH B.A

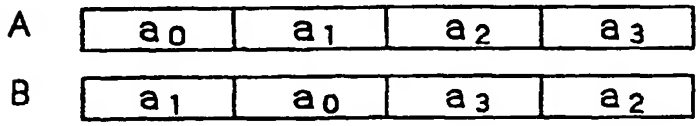

FIG.15B

PROT3 B.A.16

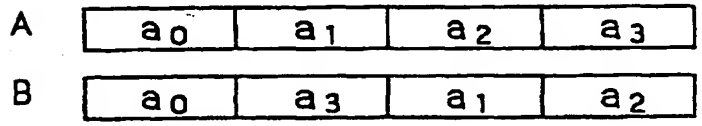

FIG.15C

PHADD B.A

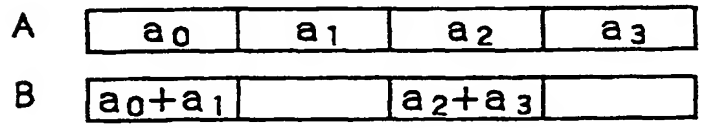

FIG.15D

PHSUB B.A

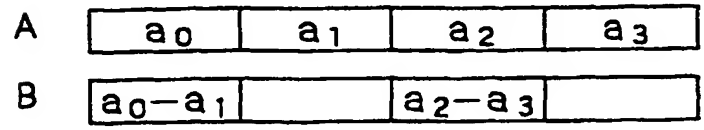

FIG.15E

BNSDOCID: <EP___0930564A1_I_>

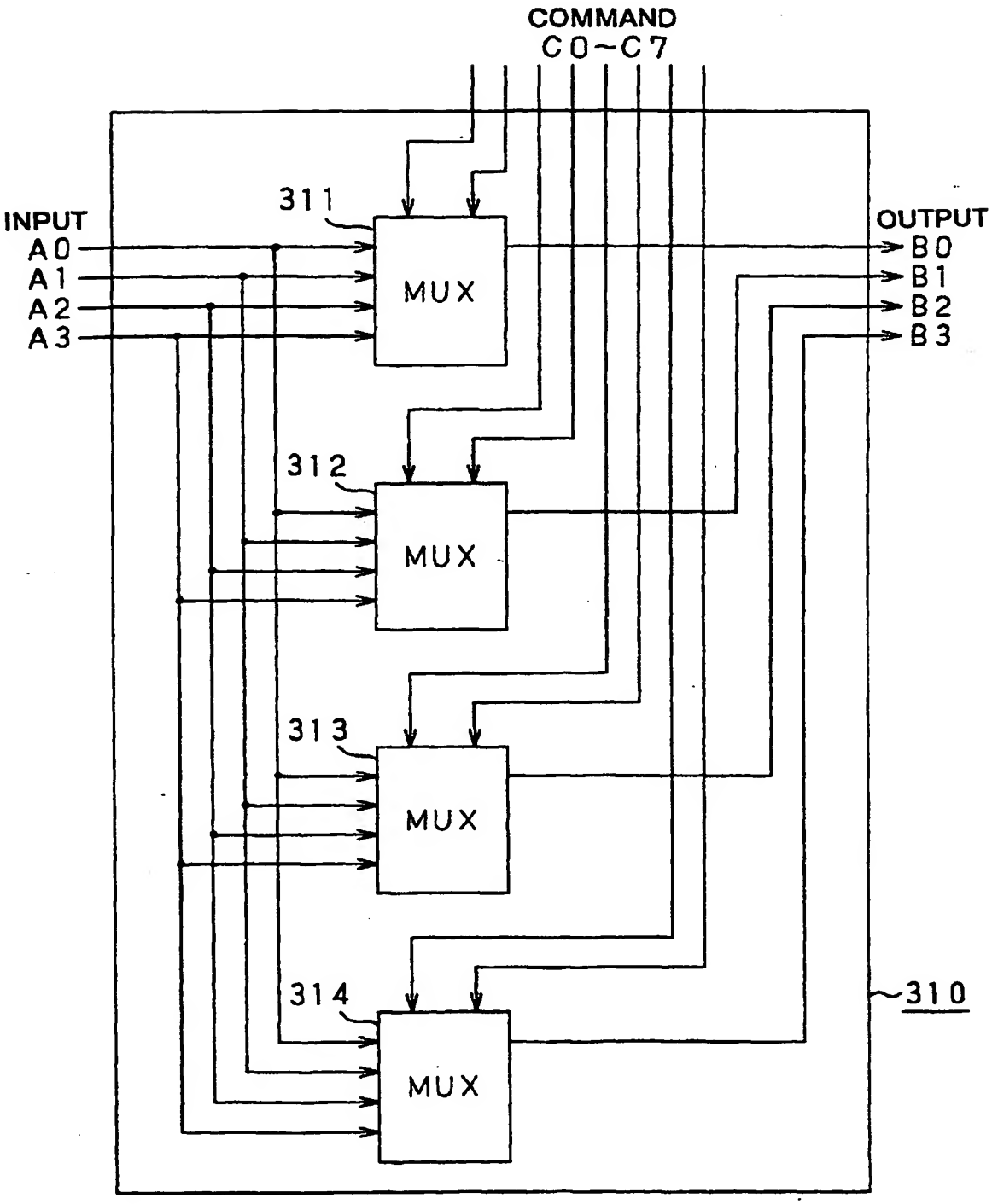

FIG.16

BNSDOCID: <EP___0930564A1_I_>

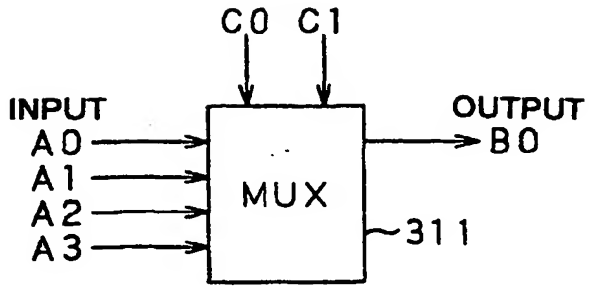

FIG.17

COMMAND C0	COMMAND C1	OUTPUT B0
0	0	A0
0	1	A1
1	0	A2
1	1	A3

FIG.18

EXC COMMAND C0	C1	C2	C3	C4	C5	C6	C7	MM INSTRUCTION
0	0	1	0	0	1	1	1	PEXC
0	1	0	0	1	1	1	0	PEXH
0	0	1	1	0	1	1	0	PROT3
0	1	0	0	1	1	1	0	PHADD
0	1	0	0	1	1	1	0	PHSUB

FIG.19

BNSDOCID: <EP___0930564A1_I_>

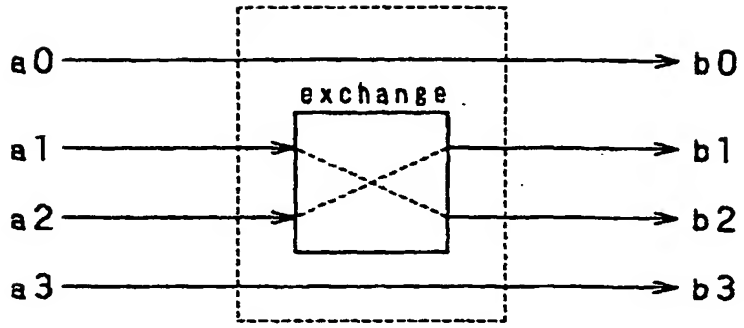

FIG.20

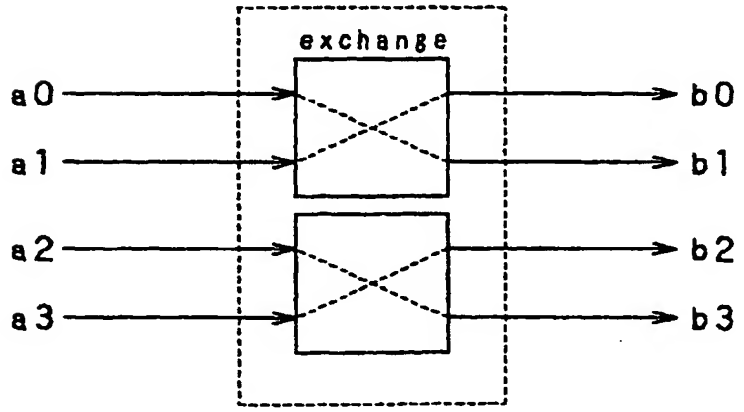

FIG.21

BNSDOCID: <EP___0930564A1_I_>

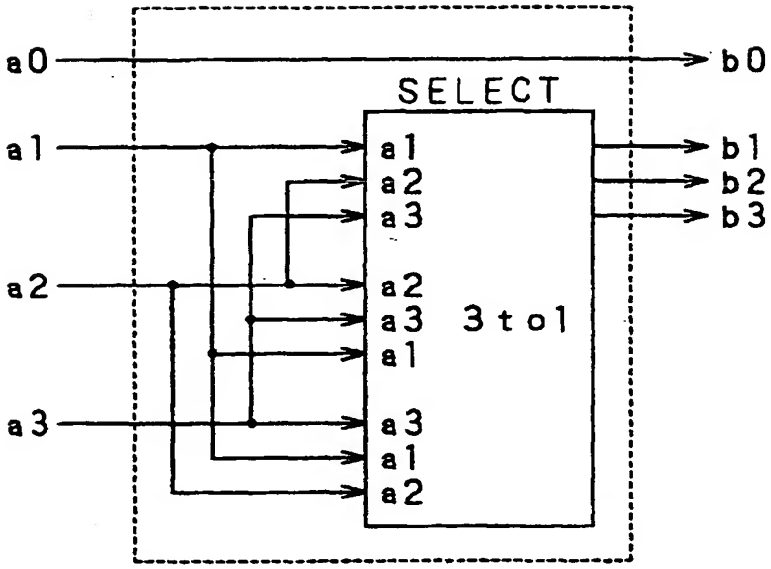

FIG.22

BNSDOCID: <EP___0930564A1_I_>

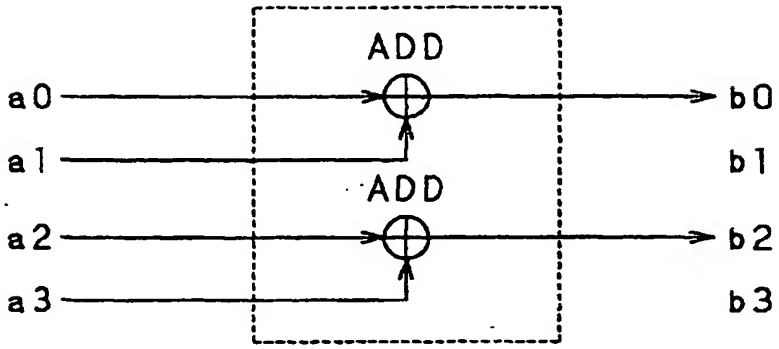

FIG.23

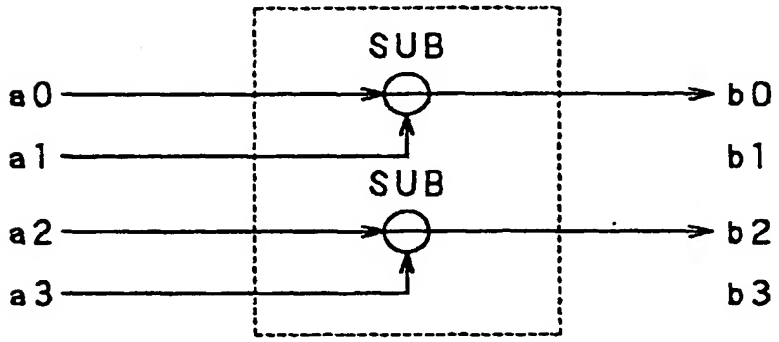

FIG.24

			63 48	47 32	31 16	15 0
		A0	0	a_{00}	a_{01}	a_{02}
		A1	0	a_{10}	a_{11}	a_{12}
PEXH	D,A1	D	a_{10}	0	a_{12}	a_{11}
PMULH	E,A0,D	E	$a_{01}*a_{12}$		$a_{02}*a_{11}$	
PSUBW	G,E	G	0		$a_{01}*a_{12}-a_{02}*a_{11}$	

FIG.25

BNSDOCID: <EP___0930564A1_I_>

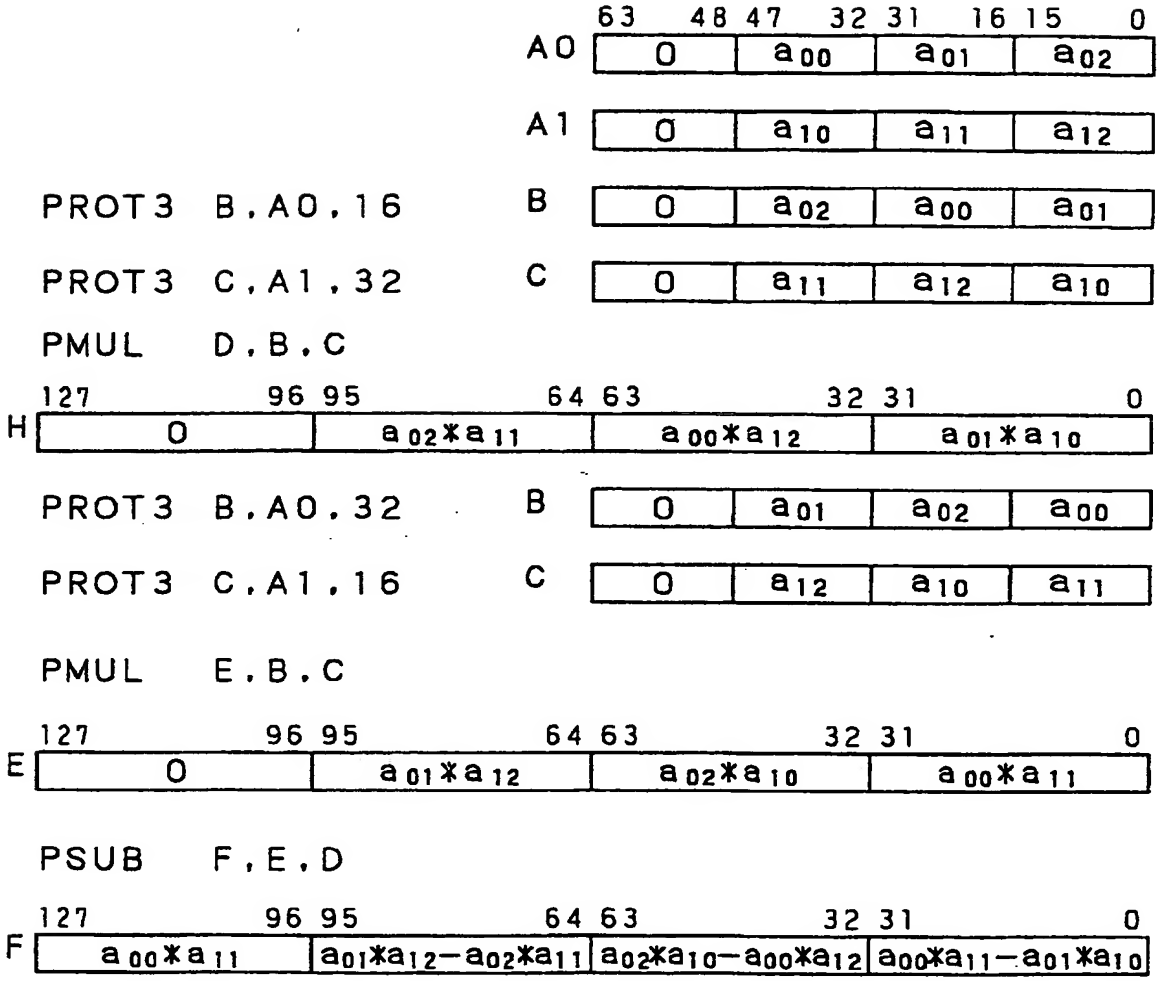

FIG.26

BNSDOCID: <EP___0930564A1_I_>

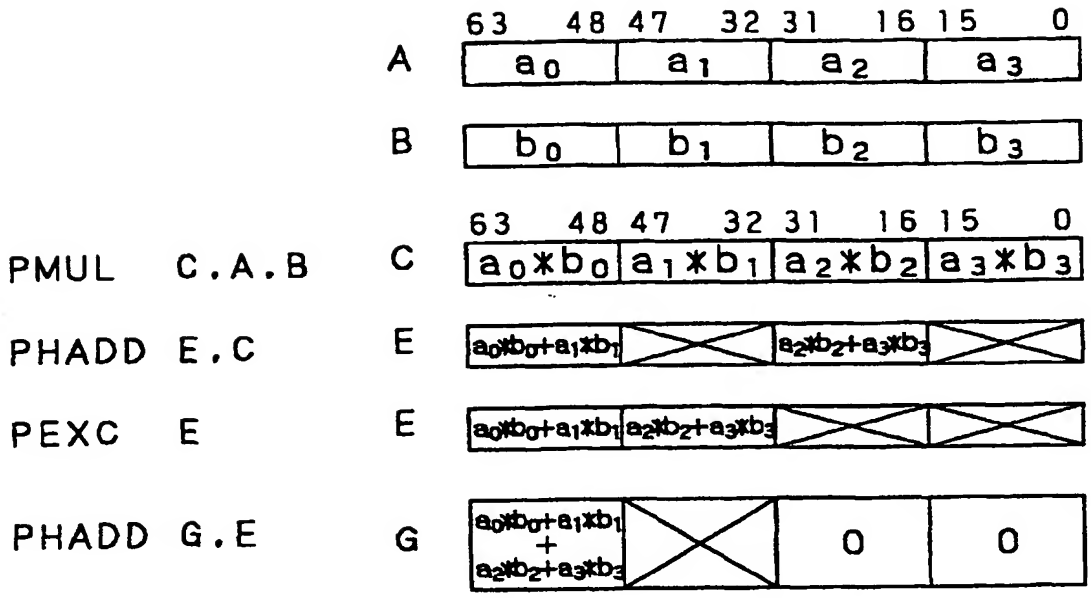

FIG.27

BNSDOCID: <EP___0930564A1_I_>

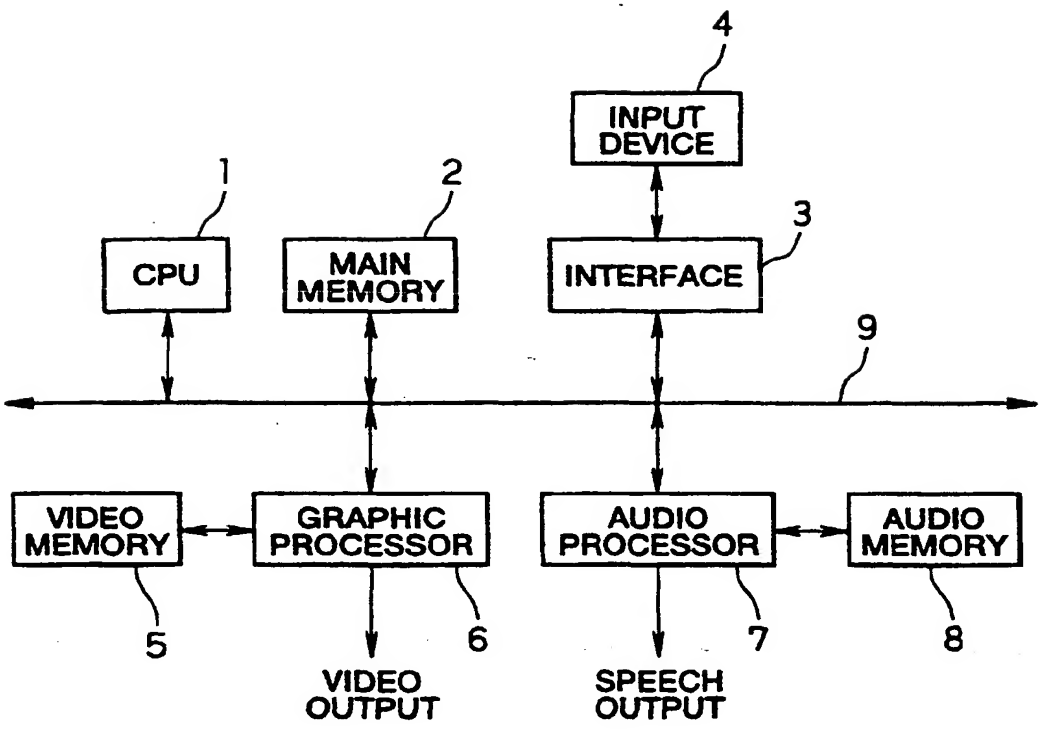

FIG.28

BNSDOCID: <EP___0930564A1_I_>

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP98/01626

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[6] G06F9/30, G06F9/38, G06F7/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[6] G06F9/30, G06F9/38, G06F7/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1926-1996 Jitsuyo Shinan Toroku Koho 1996-1998
Kokai Jitsuyo Shinan Koho 1971-1995 Toroku Jitsuyo Shinan Koho 1994-1998

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	JP, 8-328849, A (SGS-Thomson Microelectronics Ltd.), December 13, 1996 (13. 12. 96) & EP, 744686	4 1-3, 5
Y	JP, 5-241826, A (Hitachi, Ltd.), September 21, 1993 (21. 09. 93) (Family: none)	1-3, 5
A	Nikkei Electronics, No. 661, May 1996 (Tokyo) "Direction of Multi-Media Instruction MMX of 86-Series Microprocessor (in Japanese)", p.105-119	1-5
A	JP, 7-262010, A (Hitachi, Ltd.), October 13, 1995 (13. 10. 95) (Family: none)	1-5
A	JP, 9-16397, A (Hewlett-Packard Co.), January 17, 1997 (17. 01. 97) & US, 5673321, A & EP, 751456, A	1-5

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
July 7, 1998 (07. 07. 98)	July 21, 1998 (21. 07. 98)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (July 1992)

BNSDOCID: <EP___0930564A1_I_>